NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

THE DEVELOPMENT OF A STANDARD DATABASE
SYSTEM FOR REPUBLIC OF KOREA ARMY'S
PERSONNEL MANAGEMENT

by

Dae Sik Hong

June 1983

Thesis Advisor: S. H. Parry

Approved for public release; distribution unlimited

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) The Development of a Standard Database System for Republic of Korea Army's Personnel Management		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1983
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Dae Sik Hong		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1983
		13. NUMBER OF PAGES 95
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

personnel management
database management system
database system
relational database model

database
application system

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis presents the development of a database system for Republic of Korea Army's personnel management. Database processing has grown significantly in computer science areas and also in management of certain organizations. Database system designers establish objectives of database system organizations before initiating development. An important

(continued)

Item 20. (continued)

consideration in database development is to assure that it can be used for a wide variety of application and can be changed quickly and easily. ROK Army needs an end-user application system because users usually require statistical information periodically. Software engineering goals are discussed to develop an efficient end-user application system. To address softwate engineering goals, top-down design and structured programming techniques are used as tools.

The Development of a Standard Database System for
Republic of Korea Army's Personnel Management

by

Dae Sik Hong
Captain, Republic of Korea Army
B.S., Kyung Pook National University, 1975

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1983

ABSTRACT

This thesis presents the development of a database
system for Republic of Korea Army's personnel management.
Database processing has grown significantly in computer
science areas and also in management of certain organiza-
tions. Database system designers establish objectives of
database system organizations before initiating development.
An important consideration in database development is to
assure that it can be used for a wide variety of application
and can be changed quickly and easily. ROK Army needs an
end-user application system because users usually require
statistical information periodically. Software engineering
goals are discussed to develop an efficient end-user
application system. To address software engineering goals,
top-down design and structured programming technique are
used as tools.

TABLE OF CONTENTS

5

6

LIST OF FIGURES

ACKNOWLEDGEMENT

I.   INTRODUCTION

It is obvious that it is the database system era in
computer technology and applications. Database processing
has grown significantly in computer science areas and also
in management of certain organizations. The cost of
computer hardware is decreasing rapidly. As capacity
increases, the cost per bit of storage decreases. The
software cost, however, which includes development and
maintenance cost, is growing rapidly. Figure 1.1 shows
this relation [Ref. 1].

percent of cost
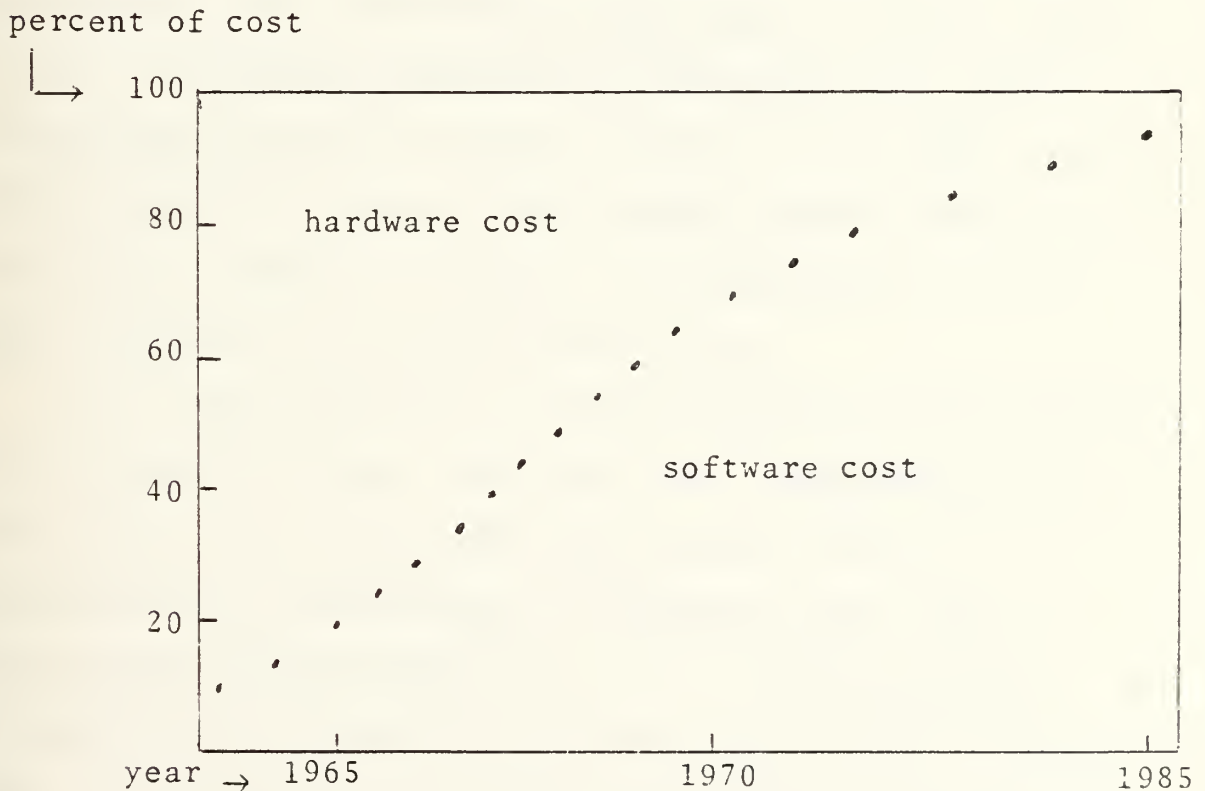
Figure 1.1.   Hardware/Software Cost Trends

11

On the other hand, personnel costs are increasing rapidly. All of the managers in any organization want to decrease the cost of production to get greater benefit. These situations motivate database system designers to continue their efforts to obtain more useful database systems.

An important consideration in database development is to store data in such a way that it can be used for a wide variety of applications and can be changed quickly and easily. To achieve the flexibility of data usage, three aspects of database design and implementation are important. First, the data should be independent of each other (data independency) and functionally dependent on the key value. Second, it should be possible to interrogate for user's requirements using application programs or the DBMS itself. Third, these data items should provide useful information for decision makers to analyze, to investigate, to plan and to manage in a certain organization.

It is very difficult to develop databases which perform in an optimal fashion. There are many different ways in which data can be structured and each has its own advantages and disadvantages. Different users want to use different data/information. It is hardly possible to satisfy all of the users with one type of data organization. The normal form concepts of relational database models will be applied to develop databases for Korean

```
┌─────────────┐
│ system      │
│ requirement │←─────┐
└─────────────┘      │
      │              │
      │     ┌─────────────┐
      └───→ │ software    │
            │ requirement │←─────┐
            └─────────────┘      │
                  │      ┌─────────────┐
                  └───→  │ preliminary │
                         │ design      │←─────┐
                         └─────────────┘      │
                               │      ┌─────────────┐
                               └───→  │ detailed    │
                                      │ design      │←─────┐
                                      └─────────────┘      │
                                            │      ┌─────────────┐
                                            └───→  │ coding and  │
                                                   │ debugging   │←─────┐
                                                   └─────────────┘      │
                                                         │      ┌─────────────┐
                                                         └───→  │ test and    │
                                                                │ validating  │←─────┐
                                                                └─────────────┘      │
                                                                      │      ┌──────────────┐
                                                                      └───→  │ operation,   │
                                                                             │ maintenance  │
                                                                             └──────────────┘
```
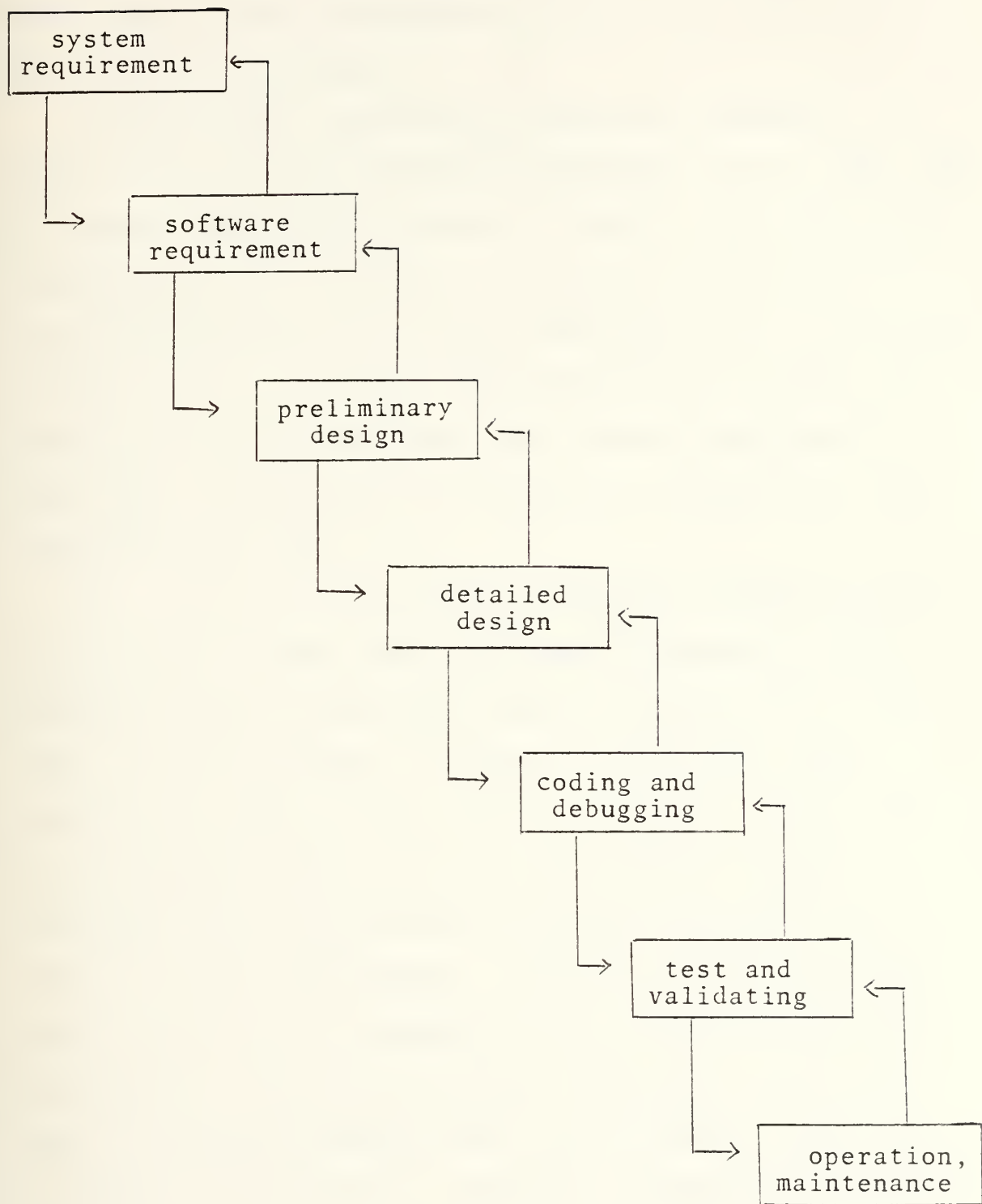
Figure 1.2. Software Life Cycle

Army's Personnel Management, because the relational data
model supports data independency better than other models
and is easy to implement.

To apply these databases for personnel management, a
commercially available database management system (DBMS) and
an end-user application system are needed. Some programming
techniques will be discussed to develop an efficient end-
user application system for any DBMS which uses a relational
database model. The basic considerations for an end-user
application system development are software engineering
goals such as modifiability, efficiency, reliability, and
understanability [Ref. 17].

As a result of this thesis, a standard database system
for Republic of Korea Army's personnel management (officer
personnel only) is developed based on software life cycle
[Ref. 2] which is shown in Figure 1.2 and other similar
publications [Ref. 13, 16, and 17]. Chapter II addresses
the background, which relates to the database system
development for Korean Army's personnel management, in
terms of system requirements. Chapter III addresses the
general overview of a database system to analyze software
requirements. Chapter IV develops databases for Korean
Army personnel management which can include all data that
is required in Chapter II. Chapter V develops an end-user
application system to extract some useful information for
personnel management from databses developed in previous

14

chapters. Finally, Chapter VI presents conclusions and recommendations based on the research presented in the thesis.

II.   BACKGROUND

A.   INTRODUCTION

The Republic of Korea (ROK) Army uses the general staff system of the USA field armies, corps and divisions, namely, G-1, personnel; G-2, intelligence; G-3, operations and training; and G-4, logistics. The Army Headquarters has the responsibility for organizing, training and equipping the ROK ground forces for the conduct of sustained combat operations. To accomplish these ground operations, the ROK Army has many service branches. The ROK Army is the largest organization among military units and also in the ROK. For national security, the ROK Army is very important because it stands face-to-face with communist North Korea on the 155 mile DMZ.

The ROK government spends a rather large percent of the total government budget for national defense, and the Department of National Defense spends a significant portion of the national defense expenditures for personnel. This is the largest investment in the ROK Army, but ROK's war power is less than communist North Korea.

In order to reduce the national defense expenditure and increase the war power, the Army needs a computerized management information system for personnel management. Therefore, some important functions of the Army's

departments of personnel management and some required
information are analyzed as system requirements, and file
systems and a database system are compared for a computerized
system selection in this chapter.

B.   FUNCTIONS OF PERSONNEL MANAGEMENT

  1.   Basic Concept of Personnel Management

    Personnel managers need data about the individual
personnel power and group personnel power to analyze, to
investigate, to plan, and to apply it for their organizations.
Information about individual personnel power can be derived
from functions involving procurement, education and
training, assignment, treatment, promotion and separation
(retirement).   Information about group personnel power
can be derived by a collection of individual personnel power.
It is important to increase individual and group personnel
power in the personnel management field so that the right
people move into the right jobs at the right times and
under the right circumstances [Ref. 5].   Individual personnel
power becomes the basis for group personnel power.   Each
factor of individual personnel management will be discussed
based on Ref. 3 and Ref. 4.

  2.   Personnel Procurement

    Personnel procurement is a process of gaining the
personnel for filling vacant positions which can not be
filled from within the organization itself.   Efficient

17

personnel procurement requires some information concerning
the candidate's education, qualification, experience,
skills, degree, health, etc. ·After candidates have been
selected, their information can be kept and maintained so
that it can be used at any time for transfer, new assignment,
promotion, etc. The various new officers procurement
organizations are described in Reference 3. These sources
and all personnel officers who are currently working in
the Army are investigated for future plans and applications.

3.   Personnel Education and Training

Information about education and training of
personnel is important and is used for personnel development
and promotion. This information is used to match or minimize
the difference between skills possessed by those who will
occupy the position. A person's educational background
can be used to gain special knowledge required for placing
a person in a particular job and to prepare that person
for a new assignment. The basic education and training
organizations which perform the education and training
to cadets and candidates are described in Ref. 4. All
Army officers must follow a specified list of education
and training courses for their development, promotion and
assignment.

4.   Personnel Assignment

Personnel assignment deals with selecting the right
officer personnel for the right positions. Three aspects
must be considered for this job.

1. Every vacant position must be filled by a person with the ability to carry out the job in the best manner.

2. The capabilities and skills of each person must be fitted to the job so that he satisfies the job area.

3. Each person who is selected for a new position must have finished compulsory education and training courses and must have carried out compulsory positions in each rank.

5. <u>Personnel Treatment</u>

Personnel treatment deals with the physical and psychological aspects of the person and the job. These include such areas as mental and physical health, recreation, rewards, personnel service, transportation, salary, retirement plans, military insurance, annual pension and vacation (periodic, sick, reward, asking, etc.), etc. Mental and physical health conditions and rewards affect promotion and new assignments. Salary, military insurance, annual pension and personnel service, etc. affect the life of the family. Recreation, awards, personnel service, transportation, retirement plans and vacations are very important for military morale [Ref. 6].

6. <u>Personnel Promotion</u>

The promotion policy is that personnel, who have finished minimum service duration in a rank and posses the capability to perform in the upper level position, be

investigated by a promotion selection committee. Therefore, this information should be prepared and provided to the decision makers, namely the promotion selection committee. In this information, the list of personnel who can be promoted should be provided according to each rank and each brand of service. The promotion point tables of all personnel should be provided by incorporating several items into these tables. These items are the career which is required in the current rank, the result of fitness reports which are prepared annually on current rank, military education, rewards and punishments, class of physical and mental health condition, and the order of promotion recommended by commanders. The promotion selection committee will select the officers to be promoted each year from officers who are recommended for promotion according to above information, and the necessary number of officers determined by the number of vacant positions. The officers to be promoted the following year are decided upon at the end of each year.

7. Personnel Separation

Personnel separation occurs when personnel voluntarily ask to be retired from the Army through the process of retirement, or when some one can not continue in the Army because of problems with their mental or physical health condition. Personnel who request retirement

must have worked for the minimum public service duration in the Army. The minimum public service durations are different between resource organizations. If personnel have studied in commissioned organizations, they must have worked double period for a commissioned duration. However, if personnel reach the age limitation, rank limitation or maximum public service duration, they must retire on that day.

Therefore, retirement information should be prepared and provided to decision makers (i.e., retirement selection committee). This information should include a list of officers who reach the maximum public service duration or a list of officers who wish to retire and have satisfied the minimum requirements, or a list of officers who can no longer work in the Army.

C. MAJOR REQUIRED INFORMATION IN PERSONNEL MANAGEMENT TO INCREASE WAR POWER

The main functions of personnel management for the ROK Army have been described. Next, what information is needed to analyze, to investigate, to plan, and to apply in those functions is described. Information personnel managers may request might include:

1. List of all new officers for each source organization concerning scholarship, classification of home town, family condition, health condition, completion rate of education and training, etc.

2. Allocation of occupation and classification of each rank based on source organization.

3. The number of cadets or candidates who should be designated in the next year or at a specified year for each source organization.

4. List allocation for each class of civilian school for each personnel rank or all personnel.

5. List allocation of all officers with each rank who have been graduated for each military education course.

6. Selection of some officers for some positions.

7. Selection of some officers in limited rank for some educational classes.

8. Summary of an officer's career from a certain previous rank up to the current rank.

9. List the data characteristics of a person or a group (military unit) of personnel.

10. Allocation of personnel service material for each rank and for each military unit.

11. Present an information list for promotion purposes for each rank and service branch, including career, result of fitness reports, education, rewards and punishment, health condition, and the order of promotion recommendation, etc.

All of the information which may be required by personnel managers can not be described, because different managers

request different information. Personnel managers might
need information for their job in addition to that described
above.

D. ADVANTAGES AND DISADVANTAGES OF A DATABASE SYSTEM OVER
    FILE SYSTEMS

All of the information required by personnel managers
are statistical in nature and must be accurate. Because it
is time-consuming to extract them, computerized systems
are needed. File systems are usually used for extraction
of some information in the ROK Army, but these have
inefficiencies which could be covered by a database system.
A database system has several important advantages over
file systems for most enterprises. These are verified
in Ref. 8 and Ref. 10 as described below.

First, the data can be shared. This reduces the time
needed to develop new systems or to respond to one-of-a-kind
requests. In effect, more information can be obtained
from existing data.

The second advantages of a database system is the
elimination or reduction of data duplication that can lead
to a lack of data integrity in conflicting reports.

The third advantage is independent creation of programs/
data. Since each application program interfaces with the
DBMS rather than directly with the database, changes to
the database may be accomodated by changes to the DBMS

without any changes to the application programs.

However, the DBMS may be expensive, typically, $100,000 or more to buy. Furthermore, the processor may occupy so much memory that additional memory must be purchased. However, these problems will occur only one time when a new database system is developed.

E. CONCLUSION

In order to increase the war power of the ROK Army, it is essential that personnel management be performed very efficiently. However, to manually manage all Army personnel is a very tedious, complex, and time consuming job. Personnel management operations will be complicated more and more. Furthermore, personnel managers and decision makers will need more accurate and more statistical information. It is impossible to get all information with a manual system or file systems which are required by personnel managers in a short period of time. Some high and middle level officers of the ROK Army are very interested in DATABASE SYSTEMS. On the other hand, the number of officers who are working in middle and high level positions must be reduced to decrease the national defense expenditure.

The database system has become an important tool for retrieving timely and accurate information and is expected to provide its user with the required information within a

specified time. Almost all of today's DBMS were developed
to manage large database systems. Therefore, a standard
database system has to be developed for efficient
personnel management in the ROK Army.

III. GENERAL OVERVIEW OF A DATABASE SYSTEM

A. INTRODUCTION

Terminology for database systems is still not standardized. Different database systems employ different words to describe the data [Ref. 7]. Therefore, confusion has arisen over the description of the database. However, it is to some extent accepted as conveying a more sophisticated concept than the older term "file". File processing systems are predecessors of database systems. They do not allow integrated processing [Ref. 8]. In order to develop a database system and to apply it, the general terminology and basic concepts must be understood by users and designers.

For the above reasons, this chapter begins with definitions of some of the basic terminologies, and then discusses architecture and user classification of a database system, and objectives of database system organizations.

B. DEFINITION OF BASIC TERMINOLOGY

1. Logical and Physical Data Description

Descriptions of data and of the relationships between data are one of two forms: logical or physical [Ref. 7, 9]. Physical data descriptions refer to the manner in which data are stored physically on the hardware, i.e., the physical database resides permanently on secondary storage devices, such as disks and tapes. However, logical data descriptions

refer to the manner in which data are represented to the application programmer or user of the data.

2. <u>Data</u>

There are many alternate words used for describing data. A widely accepted authority on databases is the CODASYL Data Base Task Group (DBTG) [Ref. 7]. Generally, data is a description of phenomena in some fashion. In the following subsections and Figure 3.1, the words used to describe the application programmer's view of data are shown.
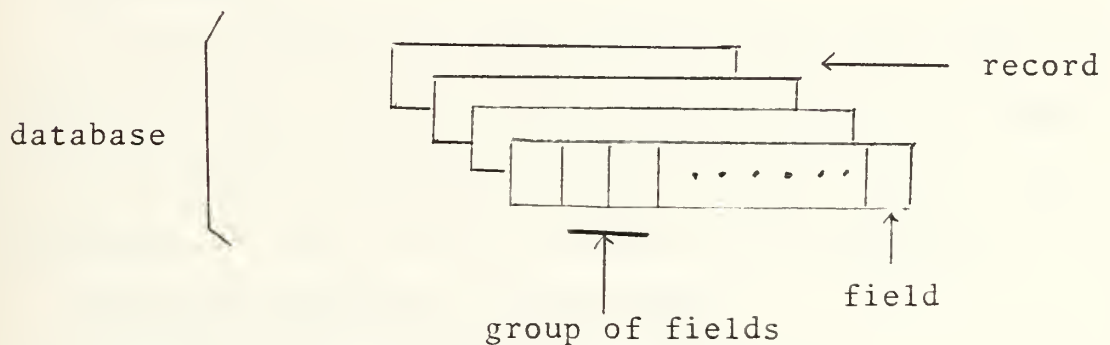
Figure 3.1. Terms Which Describe the Application Programmer's View of the Data

3. <u>Field (Data Item, Data Element)</u>

A field is the smallest unit of named data. It may consist of any number of bytes.

4. Group of Fields (Segment, Data Aggregate)

A group of fields is a collection of fields within a record which is given a name and referenced as a group. For example, a group of fields called DATE may be composed of the fields MONTH, DAY, and YEAR.

5. Record

A record is a named collection of fields or groups of fields.

6. Database

A database may be defined as a collection of records which have the same record type.

7. Databases

In most systems the term database does not refer to all the record types but to a specified collection of them. There can be several databases in one database system. The term databases is used for the collection of databases.

8. Database Management System (DBMS)

A DBMS is software that allows one or more persons to use and modify the databases. A major role of the DBMS is to allow the user to deal with the data in abstract terms, rather than as the computer stores the data. In this sense, the DBMS acts as an interpreter for a (very) high level language.

9. Database System

A database system is a combination of databases, a DBMS, and an application system (if necessary). Figure 3.2

28

shows a standard database system. The application system is
collection of end-user application programs which are
related in each view in Figure 3.3.


```
+-------------------------------------------------------+
|                                                       |
|        +---------------------------------+            |
|        |           databases             |            |
|        +---------------------------------+            |
|                         ^                              |
|                         |                              |
|                         v                              |
|        +---------------------------------+            |
|        |            DBMS                 |            |
|        +---------------------------------+            |
|                         ^                              |
|                         |                              |
|                         v                              |
|        +---------------------------------+            |
|        |      application system         |            |
|        +---------------------------------+            |
|                                                       |
+-------------------------------------------------------+
```
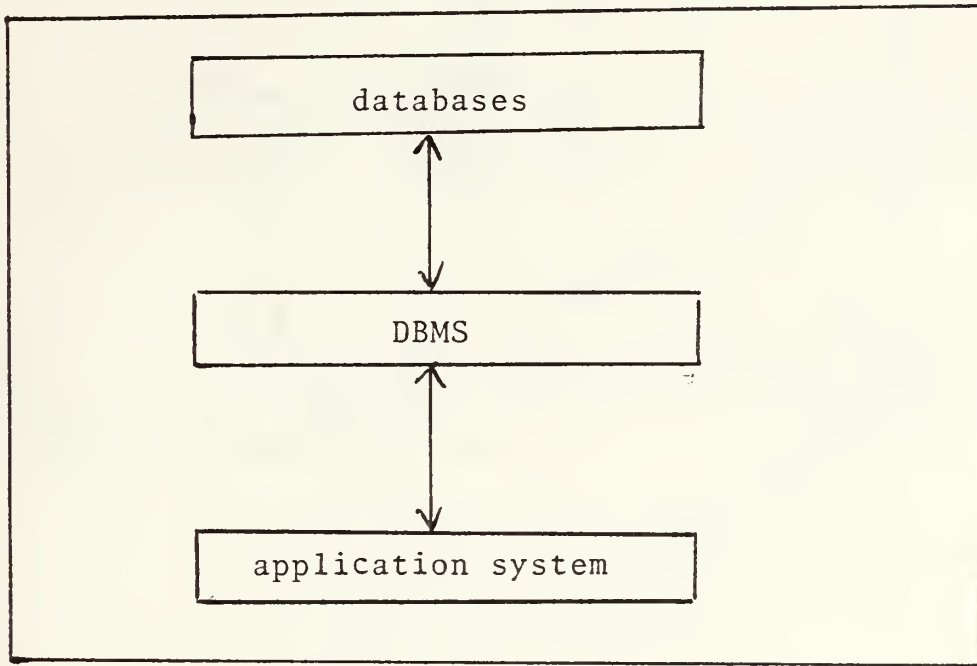
Figure 3.2.   A Standard Database
              System

C.   ARCHITECTURE FOR A DATABASE SYSTEM

The architecture is divided into three general levels:
internal; conceptual and external [Ref. 9, 10].   Figure 3.3
shows the standard view-points regarding the three levels.

In Figure 3.3 a single database, which may be one of
many databases using the same DBMS, is viewed at three
different levels.   Only the physical database exists.   The
conceptual database is an abstract representation of the
physical database, and the views are each abstractions or
portions of the conceptual database.
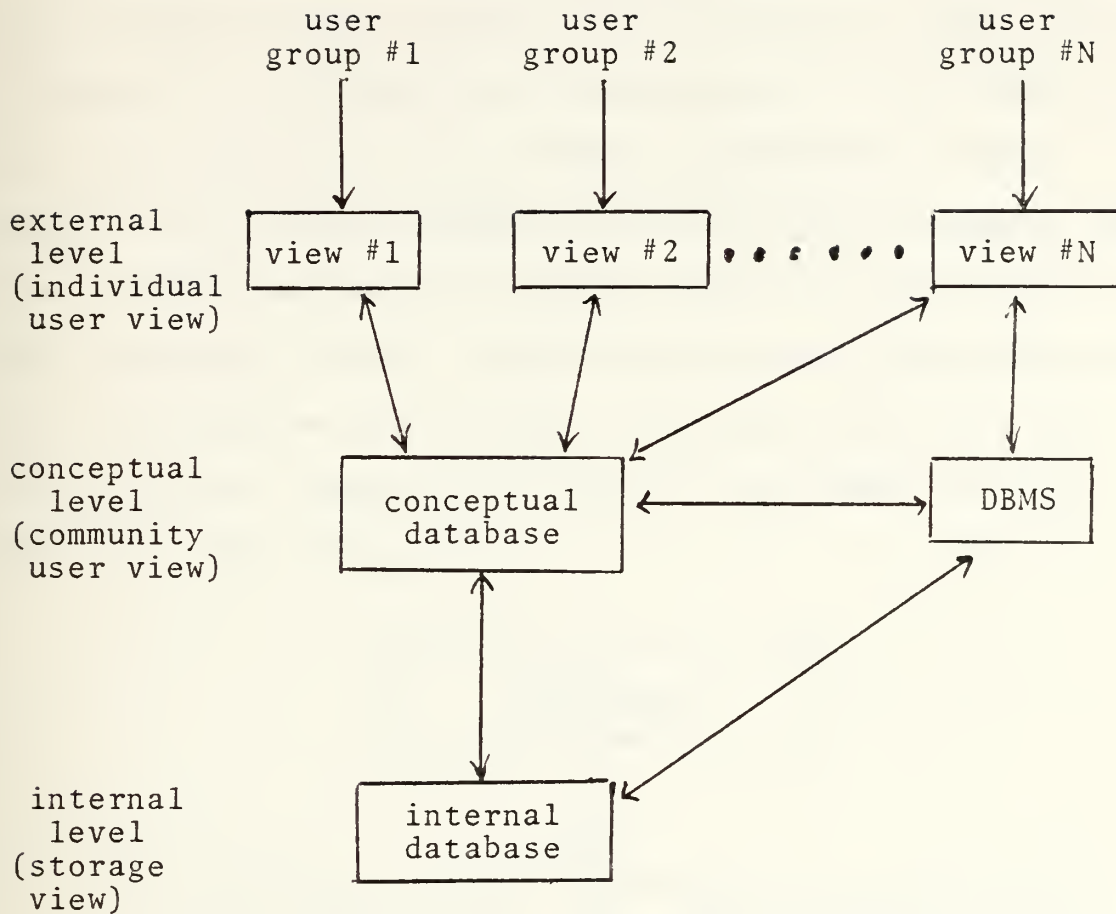
Figure 3.3.   Standard View-points of a Database System

D.   TYPICAL USER CLASSIFICATION OF DATABASE SYSTEMS

Three broad classes of users can be typically considered
[Ref. 8, 10].   First, there is the APPLICATION PROGRAMMER.
His/her responsibility is to write application programs that
use the database.   These application programs operate on
the data in all the usual ways: retrieving information,
creating new information, deleting or changing existing
information.

30

The second class of user is the END-USER. He/She accesses
the database from a terminal and may employ a query-language
(Chapter V) provided as an internal part of the DBMS. Also,
the user invokes a user-written or programmer-written
application program (Chapter V) that accepts commands from
the terminal and in turn issues requests to the DBMS on
the end-user's behalf. Either way the user may again perform
all the functions of retrieval, creation, deletion, and
modification, although retrieval is the common function
for this class of user.
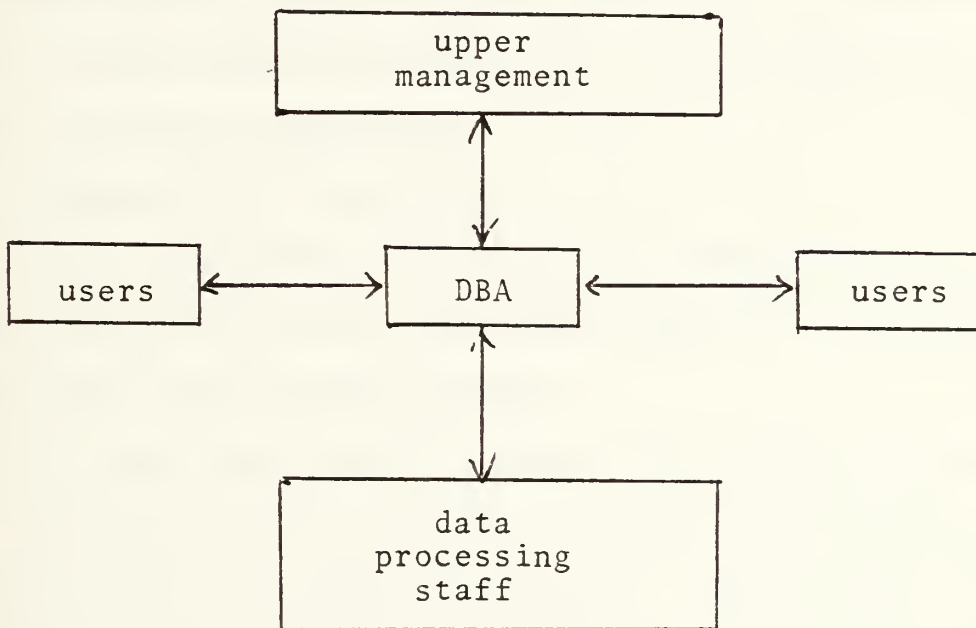
Figure 3.4.   An Example of DBA's Organizational
              Relationships

The third class of users is the DATABASE ADMINISTRATOR
(DBA). The DBA is the person (or group of persons)

31

responsible for overall control of the database system. Figure 3.4 shows an example of the DBA's organizational relationships. The DBA's major responsibilities include the following areas:

1. Deciding the information content of a database.

The DBA decides exactly what information is to be held in the database to identify the entities of interest to the enterprise and to identify the information to be recorded about those entities.

2. Deciding the storage structure and access strategy.

The DBA must decide how the data is to be represented in the database and must specify the representation by the storage structure definition.

3. Communication with users.

It is the business of the DBA to communicate with users, to ensure that the data they require is available, and to write the external schemes.

4. Defining authorization checks and validation procedures.

Since the data is a shared resource, problems occur regarding who can do what to the data. The DBA must consider each shared data component and determine access and modification rights.

5. Defining a strategy for backup and recovery.

The DBA must define and implement an appropriate recovery strategy, involving periodic dumping of the database to a backup device and the procedure for reloading the

32

relevant portions of the database from the latest backup
generation.

   6. Monitoring performance and responding to changes in
       requirements

   The DBA is responsible for organizing the system so
as to get the performance that is "best for the enterprise",
and for making the appropriate adjustments as requirements
change.

   The ROK Army has many end-users for personnel management,
but there is a lack of application programmers and DBA per-
sonnel. Therefore, the ROK Army has to identify and train
many personnel in computer technology who will work in
computer operations or computer policy areas. Furthermore,
most end-users do not have any knowledge of computer operations
and database systems. Therefore, the ROK Army has to train
personnel in computer operations who will work in personnel
management departments. Database system designers must
consider these situations.

E.   THE OBJECTIVES OF DATABASE SYSTEM ORGANIZATIONS
   A database system should be a repository of the data
needed for an organization's data processing. That data
should be accurate, private, and protected from damage.
The system should be designed so that diverse applications
with different data/information requirements can employ the
data. Different end-users have different views of data
(section D) which must be dervied from a common overall

data structure. In order to achieve these user requirements and others, the following objectives are considered by database system designers [Ref. 7].

1. THE DATABASE IS THE FOUNDATION OF FUTURE APPLICATION DEVELOPMENT. It should make application development easier, cheaper, faster, and more flexible.

2. THE DATA CAN HAVE MULTIPLE USES. Different users who perceive the same data differently can employ them in different ways.

3. CLARITY. Users can easily determine and understand what data are available to them.

4. EASE OF USE. Users can gain access to data in a simple fashion. Complexity is hidden from the users by the DBMS.

5. FLEXIBLE USAGE. The data can be used or searched in several ways with different access paths.

6. CHANGE IS EASY. The database can grow and change without interfering with established procedures for using the data.

7. LOW COST. The cost of storing and using data, and the cost of making changes, must be as small as possible.

8. LESS DATA PROLIFERATION. New application needs may be met with existing data rather than creating new files, thus avoiding the excessive proliferation in today's tape libraries.

9. PERFORMANCE. Data requests can be satisfied with speed suitable to the usage of the data.

34

10. PRIVACY. Unauthorized access to the data will be prevented. The same data should be restricted in different ways from different uses.

11. AVAILABILITY. Data should be available to users at the time when they need them.

12. RELIABILITY. Almost all information/data for personnel management is very important to both individual personnel (eg. for promotion, for new assignment, etc.) and group personnel. The information which is derived from database processing must be very reliable.

IV. DATABASE DEVELOPMENT

A. INTRODUCTION

There are many ways in which a database can be designed. Which principles should be applied in selecting a database model? Which database model should be selected to develop the most efficient database system for ROK Army personnel management? Which data items should be incorporated in a database? Which tecnique should be applied to design databases using a selected database model? The ultimate objectives of database systems organization, discussed in Chapter III, are to make application development easier, cheaper, faster and more flexible. These objectives must be considered by the database designers in designing a database system. The considerations for database model selection are ease of use, efficiency of implementation, and matching the structure of real data. The relational database model is the best database model for ease of use [Ref. 7, 9, 10, 11], even though it has some potential inefficiencies. These inefficiencies can be eliminated using query optimization discussed in Chapter V. End-users who work in the department of personnel management usually use many tables for collection of data.

B. DATABASE MODELS AND A DATABASE MODEL SELECTION

For data to be useful in providing information, they need to be organized so that they can be processed efficiently. A database model is an abstraction device. It is a pattern according to which data are logically organized. It consists of named logical units of data and expresses the relation-ships among the data as determined by the interpretation of a model of the real world [Ref. 11].

In data modeling, data must be organized so that they represent as closely as possible the real world situation. Many different models exist, such as relational, network, hierarchical, entity-relationship, binary, semantic, and the infological data model. However, only three data models are well known (relational, network, hierarchical) and are most readily available [Ref. 9, 11]. It is desirable to select one database model which is available in a commercial system to apply to the ROK Army personnel management. To select one database model among these, the main criteria by which they should be judged to achieve the objectives of a database system organization are described below.

1. Ease of Use (Simplicity) - It is felt that the less complex a data model is, the easier it is for people to understand and use it properly. The principal costs may be the time spent by the programmer writing applications programs and by the user posing queries. A model that makes accurate programming and the phrasing of queries easy and simple is desired.

37

2. Efficiency of Implementation - When databases are large, the cost of storage space and computer time may dominate the total cost of implementing a database system.

3. Matching the Structure of the Real Data - Selecting a database model involves matching the structure of real data to capabilities of the database model. For example, if the data are naturally hierarchical, a hierarchical data-base model may be the best choice for the application [Ref. 11]. Such a matching may also be desirable to minimize retraining of the users and to accommodate current data collection and entry.

Using the criteria of ease of use, there is no doubt that the relational model is superior [Ref. 7, 8, 9, 10, 11]. It provides only one representation of relationships (tables) that programmers or users must understand. Moreover, there are rich, high level languages for expressing queries on data represented by the relational model. On the other hand, the network model requires an understanding of both record types and links, and their interrelationships. Similarly, the hierarchical model requires an understanding of the use of pointers (virtual record types) and has the same problems as the network model regarding the representation of relationships. When the potential for efficient implemen-tation is considered, the network and hierarchical models score higher marks than relational models [Ref. 9].

Unfortunately, the ROK Army has many problems regarding the operation of a database system in all user groups, as described in Chapter II. Therefore, the relational database model is most appealing because of its ease of use. Even though the relational model has some inefficiencies, the storage space and computer time can be reduced using methods such as multilist, normal forms (Chapter IV), and query optimization (Chapter V). These are available in system R which was developed by IBM [Ref. 9].

On the other hand, even though individual personnel data for ROK Army personnel management is hierarchical, individual personnel data can be constructed as relational; most users need statistical information rather than individual personnel information to analyze and to plan for personnel management; and almost all data formats which are used in the ROK Army are tabular forms. These situations can be matched to the relational database model. Therefore, the relational database model appears to be the best for ROK Army database system operations.

C. STRUCTUEE OF A RELATIONAL DATABASE MODEL

The data structuring tool used by the relational database model is the relation. A relation is simply a two dimensional table. Columns of a relation are referenced as attributes. Each row of the relation is a tuple. A relation that has n columns or n attributes is said to be of degree n. Each

attribute has a domain, which is the set of values that the

attribute may have. A relation of degree n has n domains,

not all of which need be unique. To differentiate between

attributes that have the same domain, each is given a

unique identifier called an attribute name. Tuples can be

inserted, deleted, and modified in database relations

[Ref. 7, 8, 9, 10, 11].

PERSON:

RANK	SSN	MSN	NAME
captain	1928113	75-00162	Hong, dae sik
colonel	2345678	25613	Park, kil dong
major	9877655	65544	Kim, chung Su

Figure 4.1.   An Example of a Relation

In Figure 4.1, the RELATION is PERSON, and the RANK, SSN,

MSN and NAME which represent rank of person, social security

number of person, military series number of person and name

of person, respectively, are called attribute names. Each

domain of attributes can be limited using a specified number

of characters, numeric or alpha-numeric. "Hong, Dae sik"

is a value of attribute "NAME" and is included in the domain

"NAME". On the other hand, one semantic interpretation

that can be applied to a relation is to make each tuple

correspond to a particular entity. A person is an entity,
and all persons who are officers and working in the ROK Army
is an entity set in Figure 4.1.

Relational databases are specified by a relational
schema which consists of one or more relational subschemas
(scheme). A relational subschema is a listing of a relation
name and its corresponding attributes. However, a relational
schema and a relational subschema correspond to the databases
and a database in a database system, respectively. Figure 4.2
represents an example of a relational schema for ROK Army's
personnel management.

PERSON   (rank, ssn, msn, name)

FATHER   (father's name, father's ssn, msn,
          profession, address, phone #)

SCHOOL   (msn, name of school, start date, end date,
          degree)

MILITARY UNIT   (msn of an officer, unit name,
                 commander name, location)

Figure 4.2.   An Example of a Relational Schema

The specification "PERSON (rank, ssn, msn, name)" is an
example of a relational subschema, and a relational subschema
can be used to represent an entity type in relational
databse models.

Within a given relation there are one or more attributes
with values that are unique within the relation and thus can

41

be used to identify the tuples of that relations This is
called the primary key for that relation. However, not
every relation will have a single attribute primary key.
Some relations will have some combination of attributes that,
when taken together, have the unique identification property.
The model may also use a key which does not identify a unique
tuple, but which identifies all those tuples which have
that certain property. This is called a secondary key.

D.   FUNCTIONAL DEPENDENCY AND DECOMPOSITION

The major direction of most database designers effort
is to obtain an accurate schema. The concept of what is
meant by a "good"/"better" schema, and the associated condi-
tions, must be formalized.

Let X and Y be the attributes of a relational subschema,
R, and let f be a time-varying function such that f is a
function from the underlying domain of X to the underlying
domain of Y (written $f : X \longrightarrow Y$). In the precise mathematical
sense, f is not a function because it is allowed to change
over time. If f is thought of as a set of ordered pairs
$[(x,y)$ x in X, y in Y], then at every point in time, for
a given value of x in X there is at most one value of y in
Y associated with x. To distinguish f from a mathematical
function, it is called a functional dependency [Ref. 7, 9,
10, 11]. If $f : x \longrightarrow Y$, then Y is said to be functionally
dependent on X, and X is said to functionally determine Y.
$X \longarrownot\longrightarrow Y$ means that Y is not functionally dependent on X.

In order to derive a good/better schema, it is important to be able to derive all the functional dependencies, f. The inference rules (axioms) which can be used to derive implicit functional dependencies are as follows [Ref. 9, 11]:

FD1 (Reflexivity) : If Y included in X, then $X \longrightarrow Y$

FD2 (Augmentation) : If Z included in W and $X \longrightarrow Y$, then $XW \longrightarrow YZ$.

FD3 (Transitivity) : If $X \longrightarrow Y$ and $Y \longrightarrow Z$, then $X \longrightarrow Z$.

FD4 (Psuedo-transitivity) : If $X \longrightarrow Y$ and $YW \longrightarrow Z$, then $XW \longrightarrow Z$.

FD5 (Decomposition) : If $X \longrightarrow YZ$, then $X \longrightarrow Y$ and $X \longrightarrow Z$.

(where W, X, Y, and Z are subsets of the attributes)

Generation of all functional dependencies is very time consuming, because usually there are many functional dependencies in a subschema. Functional dependencies can be used to evaluate the schema and to decompose it into a better schema. Other reasons have also been suggested why decompositions are necessary [Ref. 9, 11]. In Figure 4.2, the relational subschema

MILITARY-UNIT (msn of an officer, unit name, commander name, location)

may have the following anomalies and redundancy in manipulation:

1. Update Anomaly - The change of commander name in a military unit necessitates a series of changes of commander's name.

43

2.   Insertion Anomaly - When an officer is assigned to
a military unit, the commander's name and location of unit
must be included.

3.   Deletion Anomaly - When an officer is transferred to
another military unit, any military unit information will
cease to exist. This can be considered an anomaly if it
is desired to retain important, long-range information
about the unit.

4.   Redundancy - The commander's name and location of unit
are repeated in many tuples. This redundancy causes
problems, not only because it is wasteful (storage), but
also because redundant data must be consistently maintained.

These problems can be avoided by decomposition. In the
above example, the anomalies can be eliminated by breaking
the relational subschema into two relational subschemas:

PERSONNEL-ALLOCATION (msn, unit name)

MILITARY-UNIT (unit name, commander name, location)
In the decomposed subschema, PERSONNEL-ALLOCATION and
MILITARY-UNIT are isolated and related only by specifying
the unit name in which the officer works. This decomposition
is not arbitrary. It is based on the two functional
dependencies msn $\longrightarrow$ unit name, and unit name $\longrightarrow$ commander name,
location. The decomposition isolates these two dependencies
in separate relational subschemas. As a result, they do
not interface with each other. In addition, the separate
subschemas are considered better than the original subschema

when the natural join (described in Chapter V) of two subschemas is equivalent to the original subschema. Therefore, all of the rules (axioms) and properties above will be used to derive a "good"/"better" schema in the following sections.

E. SCHEMA DESIGN

A relational schema using all of the results and theories which are derived from previous sections, and using the functions of departments of personnel management are designed in this section.

1. Requirements Analysis

The first step of schema design is requirement analysis. This step consists of a high-level analysis of the function of an organization (i.e., departments of personnel management in the ROK Army). The functions of departments of personnel management and required information were given in Chapter II. The purpose of this step is to:

a. Gain familiarity with the area of the organization to be modeled.

b. Determine the information requirements of the organization without regard to constraints other than the way in which an organization does business, and

c. Represent these requirements via some formal modeling technique.

To gain familiarity with the organizational area, the organization must be understood in terms of its goals and the strategies it uses to achieve these goals. To determine

45

the information requirements, data on processes in the
organization must be collected. Finally, data modeling
techniques are used to formally represent the information
requirements.

Information requirements are collected from users
at all levels in the organization. In most organizations
at least three levels that provide data can be identified:
top management, middle management, and operations management
[Ref. 11, 12]. From top management, information on the goals
and objectives of the organization can be obtained, along
with strategies and methods for managing the implementation
of the strategies. Middle management provides more detailed
policies and constraints, and provides data about required
response time, reliability, security, and privacy, etc.
Finally, operations management provides more specific infor-
mation, such as names, sizes, number of occurrences, integrity
constraints, reliability, security and privacy of data.
They also provide information on data usage, volume, frequency
of occurrence of transactions, priority of transactions and
sequencing with other transactions.

The main purpose of this step is to understand users'
need. Subsequent steps of the schema design process can
transform these needs to subschemas according to the
relational data model. This approach to requirements analysis
produces a business model (Appendix A) as an outcome of this
step.

2. Integration of Required Data Classes

In Appendix A, each data class in a function represents a view of the subschema/schema. These views are then integrated to form a subschema which shall be integrated to form an enterprise description which describes the entire schema. This description (Appendix E) is used primarily for communication between the users and the schema designers.

Different users in different functional areas usually require different data (attributes) in each data class. Thus, common entity types must be extracted from the data classes through the integration of data in all functional areas for each data class. The following questions must be answered regarding the entity types

a. What are the entity types described by each data class in a functional area?

b. What is the appropriate name for each entity type?

c. What is the meaning of each entity type?

d. What attributes are of interest for each entity type?

e. What is the appropriate name for each atribute?

f. What is the meaning of each attribute?

Based on these questions, entity types have been specified for each data class (Appendix B). An entity type defines what it represents and specifies its associated attributes.

Entity type identification is an iterative process. The description of an entity type may change many times

before everyone agrees that it is right, or may change during the analysis of normal forms or design of the data dictionary (Appendix E) which is the final documentation for each entity type.

3. Analysis and Design of Normal Forms

Functional dependency and its axioms, decomposition, and original entity types were described in previous sections and in Appendix B. The original entity types have many anomalies which should be eliminated to realize an acceptable schema. Anomalies are caused by certain unwanted functional dependency structures. These structures can be avoided by forcing some restrictions on the allowed functional dependencies in a relational subschema.

Relational subschemas that are satisfied by the restrictions are said to be in normal forms [Ref. 7, 8, 9, 10, 11]. Though there are five normal forms, only three will be considered. Even though the others can eliminate redundancy, it becomes too complex to manipulate the database system derived from five normal forms and increases the execution time. A relational subschema is converted into a normal form relational subschema by decomposition.

Consider a relational subschema, R, and a set of functional dependencies, F. A superkey, X, of R is a set of attributes, X, of R such that for every attribute, A of R, $X \longrightarrow A$. A key of R is a superkey which is nonredundant. An attribute of R is prime if it participates in a key.

Since a relation can have many keys, they are sometimes called candidate keys. An attribute or a collection of attributes, B, can be said to be fully functionally dependent on another collection of attributes, A, if B is functionally dependent on all, as opposed to part A. Finally, attribute C is transitively dependent on attribute A if there is an attribute B such that: $A \longrightarrow B$, $B \longrightarrow C$ and $B \longrightarrow\!\!\!\!\!/ \longrightarrow A$.

Relation, R, is in first normal form if every attribute is a simple attribute. That is, there are no composite attributes in R.

Relation, R, is in second normal form if it is in first normal form and every nonprime attribute of R is fully functionally dependent on the keys.

Relation, R, is in third normal form if it is in second normal form and it has no transitive dependencies among nonprime attributes.

Even though the three normal forms are analyzed and designed, only the entity types which are designed with third normal form are shown in Appendix C. This job is time consuming and difficult, but is a very important step in database design to accomplish the objectives of database system organizations efficiently and to reduce some inefficiencies in relational database models. Some subschemas which can be manipulated by a manipulation language (Chapter V) can be eliminated, and some subschemas which have almost the same attributes and exactly the same

49

functional dependencies can be combined into a single subschema through the normal forms design and analysis. These phenomena are discussed in Appendix C.

4. Analysis and Design of Relationships Between Entity Types

The result of the normal form design step is a list of entity types, entity names and their attributes. If some complex information from a database system is required, relationships are needed. To help identify relationships between entity types, the following questions are posed.

1. For each function, what are the known correspondences (relationships) between entity types associated with the function?

2. What is the appropriate name for each relationship type?

3. Is the relationship type expressible in closed form using the attributes of the entity types (e.g., is PERSON HAS BEEN PROMOTED (in Appendix D) true when military serial number in PERSON equals military serial number in PROMOTION LIST)?

4. What is the meaning (semantics) of each relationship?

As a result, there are many relationships between two or more entity types in each function. Some sample relationships are shown in Appendix D. Fortunately, all of the relationships are expressible in closed form using

attributes of the entity types (e.g., military serial number, rank, branch, ...). Relationships are not grouped for each function in Appendix D, because they overlap with many functions and data classes as shown for each function in the business model (Appendix A).

F.   CONSTRAINTS ANALYSIS AND ASSIGNMENT

Finally, to complete the database design step, the constraints on the attributes, entity types, and relation-ships must be identified. Constraints must be subjected to user scrutiny. Constraints are identified by asking questions such as:

1.   What is the domain of values for each attribute (e.g., is military serial number between a 5 and 9 digit number)?

2.   What are the known functional dependencies between attributes of each entity type?

3.   What are the keys for each entity type?

4.   What is the mapping property of each relationship (e.g., one to one, one to many, many to many)?

Some of the constraints (functional dependencies and keys) are identified in Appendix C through normal form analysis. The mapping property of each relationship type is included in Appendix D, and domain constraints are included in the data dictionary (Appendix E).

It is difficult to arrive at a set of constraints that represent the application and is consistent and feasible,

because some forms of the constraints are difficult to under-
stand are prone to misunderstanding and errors. There-
fore, they must be reevaluated during the database test
step.

G.   DATABASE TESTING

The testing stage may require up to half of the total
effort. Testing may be required both in each development
step and before databases are implemented. Inadequately
planned testing often results in woefully late deliveries.
A domain is the set of permissible real data to a database,
and a test is a subset of the domain.

A testing criterion specifies what is to be tested. A
general testing criterion may include:

1. Review all of the questions and analyses in the
development process.

2. Compare the data dictionary and the capacities of
the object DBMS.

A testing is complete if the test meets all the require-
ments of the test criterion, and databases are reliable if
every discovered error is revealed by each complete test
[Ref. 13]. Reliability may be achieved in three ways:
walkthrough, documentations reading, and error checking
programs.

A walkthrough is a user review to discover errors in
the data dictionary and subordinate documentation. This

is scheduled periodically for all users in each development step and is intended to detect errors, not to correct them. A very effective version of the walkthrough is documentation reading. Second designers review the documentation, including the data dictionary. This technique frequently turns up errors when the second reader, failing to understand some aspects of the documentation, asks the designer for an explanation.

Error checking programs are a part of the end-user application system to enhance reliability and validity during the creation, insertion, and modification in a specified database. These programs check values of each attribute which has limited values (e.g., attribute RANK has limited values), and notify users of errors when users create a new database, insert records into a database, or modify some fields in a record. These programs will be included in an end-user application system in the next chapter.

H. CONCLUSION

Conceptual databases of a component of a database system developed for ROK Army's personnel management are described in this chapter.

The database system organization objectives provide a guide for all of the database development steps, including selection of a data model and database design technique. The relational database model was selected to develop the most useful database system in the ROK Army environment.

The business model was used to gain familiarity with the organizational area and to integrate similar attributes into a data class in many different functional areas. The normal forms (1st, 2nd, 3rd), those based on functional dependencies, are applied to reduce the inefficiency of the relational data model in storage space and to eliminate anomalies and redundancy. The other normal forms (4th, 5th) were not applied in order to maintain simplicity during the complex information extraction. Relationships, which are expressible in closed form using the attributes of the entity types, and constraints are developed in order to help user understanding. Finally, a database description (data dictionary) which communicates between database designers and users has been developed.

The documentation, including the data dictionary, must be tested in each database development step before the database is operational. This database development method is a technique to achieve the objectives of database system organizations for ROK Army's personnel management. This technique enhances simplicity of the process to database designers. Users may understand the database design steps and database itself very easily. Furthermore, users may manipulate the database by themselves.

V.   END-USER APPLICATION SYSTEM DEVELOPMENT

A.   INTRODUCTION

Up to now, the main functions of the ROK Army's departments of personnel management and some important theories which are relevant to database system development have been discussed. Also, databases based on the relational database model have been described. In this chapter, an efficient end-user application system for any DBMS which uses a relational database model is discussed.

Query languages or end-user application programs are used to extract information from the databases. The query languages for the relational data model usually break down into two broad classes: algebraic languages and predicate calculus languages. These abstract query languages are not implemented exactly in any existing DBMS, but they serve as a benchmark for evaluating existing systems. Since each of the two abstract query languages is equivalent in expressive power to the other [Ref. 9], only the algebraic query language will be discussed in this chapter. A real query language provides the usual capabilities of the abstract languages, as well as additional capabilities.

The end-user application programs are a collection of query languages and resemble a general program using a very high level language to extract complex information or make

frequent/periodical queries (daily, weekly, monthly,...).
As discussed in Chapter IV, the relational database model
has some inefficiency of implementation in storage and
execution time. The inefficiency should be reduced by
database system designers as much as possible. The database
has been designed using normal forms to eliminate the
anomaly and redundancy. Also, designers must attempt to
reduce execution time using techniques such as query
optimization.

In order to achieve the objectives of the database
systems organization, software engineering goals — under-
standability, reliability, efficiency, modifiability —
will be considered. To address software engineering goals,
top-down design methodology and structured programming will
be applied as tools.

B.   CAPABILITIES OF AN ABSTRACT ALGEBRAIC QUERY LANGUAGE

There are five basic operations, as well as a few
additional operations, that may add to the set of operations
in the algebraic query languages. The five basic operations
are:

1.   Union - The union of relation R and S, denoted R U S,
is the set of tuples that are in R or S or both. The union
operation is applied only to relations of the same degree.

2.   Set Different - The difference of relations R and S,
denoted R - S, is the set of tuples in R but not in S.
R and S must be of the same degree.

3. Cartesian Product - Let R and S be relations of degree
r and s, respectively. The Cartesian product of R and S,
denoted R x S, is the set of (r + s) - attributes whose first
r attributes are from a tuple in R and whose last s attributes
are from a tuple in s.

4. Projection - Let the relation R have four attributes,
noted by R (aa, bb, cc, dd). Then the projection aa and bb
of R, denoted R [aa, bb], means remove attributes cc and dd,
and rearrange the remaining attributes aa and bb to form
a new relation, S (aa, bb).

5. Selection - Let F be a formula involving (i) operands
that are constant or attributes (ii) the arithmetic comparison
operators <, =, >, <=, >=, and /=, (iii) and the logical
operators "and", "or" and "not". The selection of F, denoted
$S_F(R)$, is the set of tuples in R which satisfies the formula F.

Some additional operations are intersection, quotient,
join and natural join in abstract algebraic query languages.
The join is the composition of Cartesian product and selection
and may require excessive time for execution. The natural
join, denoted (R X S), is important to prove that two
or more decomposed subschemas are equivalent to the original
subschema, as discussed in the previous chapter. To computer
(R |X| S), first compute (R X S). Next, for each attribute
A that is named in both R and S, select from (R X S) those
tuples whose values agree in both R.A and S.A. Finally, for
each attribute A, project out the column S.A. Figure 5.1

msn	unit-name
75-00162	2xx
225613	1xx
654443	3xx
34567	2xx
987654	1xx
44444	2xx

Relation: PERSONNEL ALLOCATION

unit-name	commander name	location
1xx	Hong, dae sik	Seoul
2xx	Park, kil dong	Pusan
3xx	Kim, chung su	Taegu

Relation: MILITARY UNIT

msn	unit name	commander name	location
225613	1xx	Park, kil dong	Pusan
987654	1xx	Park, kil dong	Pusan
75-00162	2xx	Hong, dae sik	Seoul
34567	2xx	Hong, dae sik	Seoul
44444	2xx	Hong, dae sik	Seoul
654443	3xx	Kim, chung su	Taegu

Relation: PERSONNEL ALLOCATION |X| MILITARY UNIT

Figure 5.1.   Operation of Natural Join

shows the operation of natural join using the example of
decomposition which is given in the previous chapter.

C.   DATA MANIPULATION LANGUAGE

The operations of algebraic query languages were
previously discussed. The notation for expressing queries
is usually the most significant part of a data manipulation
language. Data manipulation languages usually have operations
beyond those of query languages. Of course, all data
manipulation languages include insertion, deletion, and
modification commands, which are not part of the query
languages. Some additional operations are frequently available
such as arithmetic, assignment and print commands, aggregation
of function (eg. average, sum, total, min, max,...) and so
on.

D.   QUERY OPTIMIZATION

High level query languages allow the writing of queries
that may take a great deal of time to execute. Execution
time can be significantly reduced if the query language
processor rephrases the query before executing it. Such
improvements are commonly called optimization. Programmers
might ask why certain queries take a long time to execute.
The greatest offender for query languages based on the
relational model is the query that involves a Cartesian
product, join or natural join. If programmers apply some
strategies to a given query, they can reduce the execution
time. The general strategies are [Ref. 9]:

1. Perform selection as early as possible.

2. A Cartesian product followed by a selection.

3. Look for common subexpressions in an expression.

4. Cascade selections and projections.

5. Combine projects with a binary operation that preceeds or follows it.

6. Combine certain selections with a prior Cartesian product to make a join.

Programmers should attempt to move selections and projections as far down the parse tree of the expression as they can, although they want a cascade of these operations to be organized into one selection followed by one projection. They should also group selections and projections with the preceding binary operation, such as union, Cartesian product, or set difference, where possible.

E. TRANSACTION PROCESSING ANALYSIS

In order to design application programs, designers have to analyze the transaction processing of each organizational area with respect to the database. This analysis specifies inputs and outputs required, but does not involve the specific steps to obtain the output. All current and projected transactions are included. For each trasaction designers identify its nature (e.g., retrieval, update), its frequency, its origin (functional area), and its purpose.

To help identify requirements for supporting transactions, some of the relevant questions to ask are:

1. What transactions are required by each functional area?

2. What entity types and relationships are involved in each transaction?

3. What kind of access is required by each transaction (e.g., retrieval, update)?

4. What is the frequency of each transaction (e.g., daily, weekly, monthly,...)?

5. What is the processing priority of each transaction?

6. What reports are needed?

7. What is the format of each report?

8. What security requirements are important?

The result of this analysis is a list of all transactions. A sample list of some transactions, which are processed by users in ROK Army's departments of personnel management, is given in Appendix F. The list, because of its usefulness for design of an end-user application system, will be used in the next section.

F. SYSTEM DESIGN

1. General Design Concept

In the end-users application system design stage, the algorithms are developed, and the overall structure of the database system takes shape. The application system must be divided into small parts (modules), each of which is the responsibility of an individual or a small organizational area. Each such module must have its function and purpose defined. As submodules are specified, they are represented in a tree diagram showing the nesting of the system's components.

Because the solution may not be known when the design stage begins, decomposition into small modules and submodules may be quite difficult. A common problem is that the designer of an application system often does not know exactly what end-users want. Each transaction processing analysis may guide the solution to this problem. In this stage, software engineering goals must be considered in order to achieve the objectives of the database system organizations such as low cost, simplicity (comprehensibility), high performance, privacy and reliability.

Top-down design can be related to structured programming, which has been erroneously called "gotoless" programming [Ref. 13], which is more efficient than "goto" programming [Ref. 14 and 15]. Top-down design has been proposed as a methodology for reducing the complexity of design. The goal of top-down design is to minimize logical errors and inconsistencies through structural specification of the development process. System design consists of a sequence of a refinement steps. Therefore, although other design methodologies could be used, top-down design shall be applied.

    2. Preliminary Design

        In the top-down design, a subroutine is first formulated as a single statement, which is then expanded into one or more of the data manipulation language statements described in a previous section.

As a result of the preliminary design step, Figure 5.2 shows a baseline hierarchical diagram of an application system for end-users in the ROK Army's departments of personnel management which includes an error checking program previously described. Each transaction process described in the previous section affects the selection of levels and modules in the baseline diagram which is based on functions of each operational area.

Since different users need different information, application programmers have to modify the existing application programs or append new application programs to the existing system for any specified functions and purposes when the database system is operating.

Since database security is very important in the military, the DBA can include authorization check statements in each module. Furthermore, the DBA can store frequently used databases on high speed secondary storage. Therefore, this hierarchy of the application system should result in an increase in security, simplicity, modifiability, performance, and reliability.

3. Detailed Design

For each level and module, each statement which is written with the abstract algebraic query language is expanded in increasingly greater detail until the resulting description becomes the actual source language program in a selected DBMS. In order to reduce the execution time,
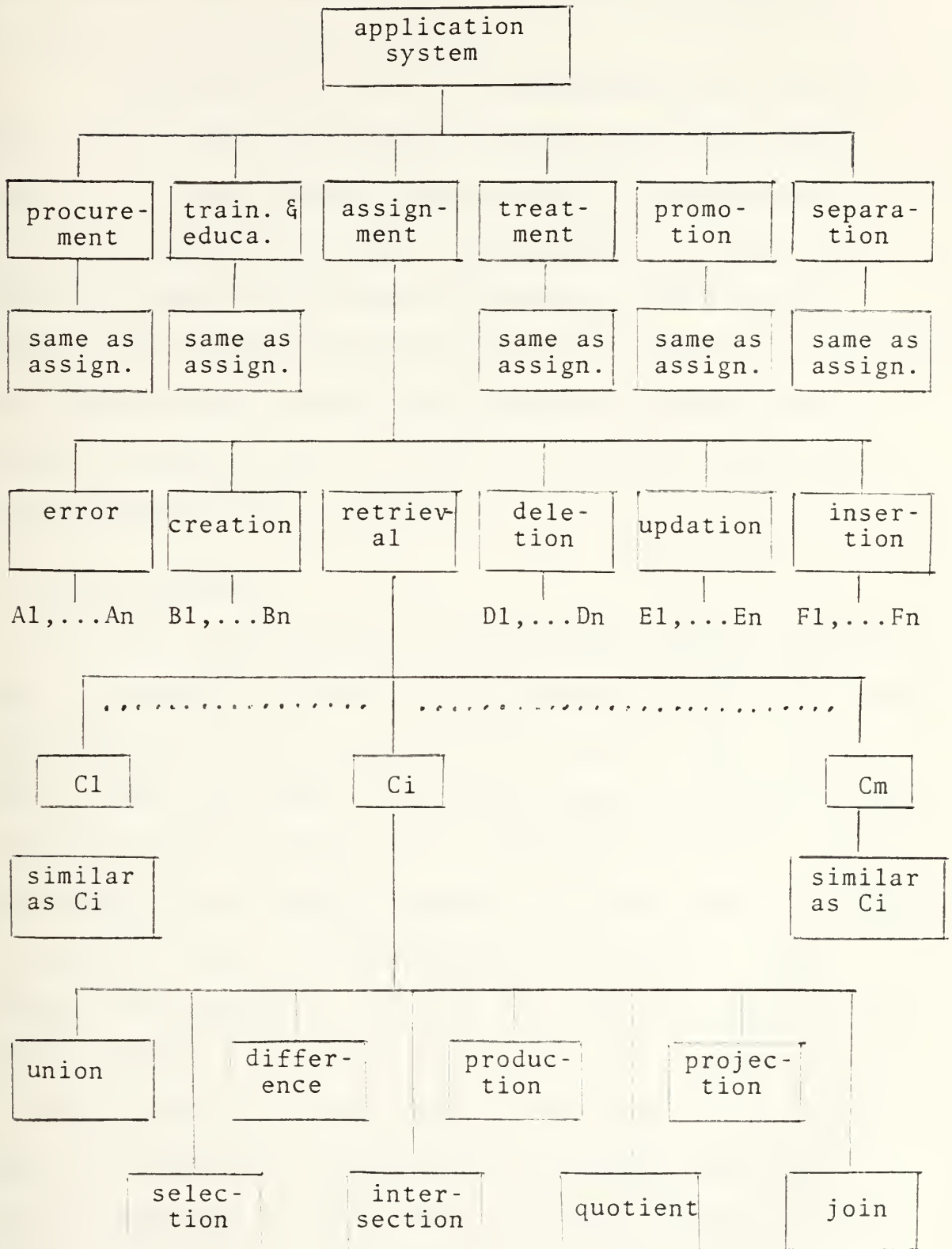
Figure 5.2. A Baseline Diagram for an Application System

the designer must consider query optimization before the actual source language is applied.

The premise of structured programming is to use a small set of simple control structures with simple proof rules. A written program then is built by nesting these statements within each other. This method restricts the number of connections between program parts and thereby improves the comprehensibility and reliability of the program. The if-then-else, while-do, and sequence statements are a commonly suggested set of control structures for this type of programming [Ref. 13 and 16].

G. SYSTEM TESTING

The basic requirement for application systems is correctness. A program is correct if the output satisfies the output requirements for every input dictated by the input requirements. Since the number of possible inputs is usually large, checking program correctness by examining the program inputs and outputs is not always feasible. In most cases, a program is tested by a set of "representative" inputs. If the outputs corresponding to this set of inputs are correct, the program becomes operational and is released for general use.

Since testing can only indicate the presence of certain types of errors, it is impossible to estimate the number of errors remaining in the system. Testing can not guarantee program correctness. For these reasons, the reliability

65

a program is defined as the probability that the next invocation of the program is correct. A program which is correct with respect to tests performed on a few selected relations can be considered extremely reliable if the inputs tested constitute 99.9 percent of all inputs to the program [Ref. 16]. It may be impossible to apply this definition to the very large databases. Thus, Ref. 16 suggests top-down design and structured programming as software development tools, and these have been applied already.

Well organized system testing can improve the reliability of the program and improve user confidence, which are the objectives of most testing. In order to improve the reliability and user confidence, the test plan should be designed early and most of the data should be specified during the design state. Testing is divided into three distinct operations [Ref. 13].

1. Module Testing subjects each module to the test data supplied by the programmer.

2. Integration Testing tests groups of components together. Eventually, this procedure produces a completely tested system. This testing frequently reveals error missed in module tests.

3. System Testing involves the test of the completed system by an outside group. The independence of this group is important.

In each operation, the programmer should check that:

1. Every statement has been executed at least once by the test data.

2. Every path through the program has been executed at least once by the test data.

3. For each specification of the program, test data is used to determine whether the program performs the particular specification correctly.

H.   CONCLUSION

A end-user application system of a component of a database system developed for ROK Army's personnel management are described in this chapter.   The end-user application system includes many application programs which are required for extraction of users' required information, creation of new databases, or deletion and updation of existing databases. Transaction processing of each functional area has been analyzed before the system is designed in order to help the designer's understanding.   This analysis specifies what the system is to do, but does not include how the system is to do it.

The software engineering goals are considered to achieve the objectives of database system organizations.   To address software engineering goals, top-down design methodology, and structured programming technique are applied as tools, and a baseline hierarchical diagram of the system is developed.

The premise of structured programming is to use a small set of simple control structures. Abstract algebraic query languages are applied in the preliminary design step. Query optimization techniques must be considered before actual source language is applied in the detailed design step in order to reduce execution time, which is an inefficient element in the relational database models.

System testing is a tremendous job. The top-down design methodology and structured programming technique can decrease the amount of testing because they restrict the number of connections between program parts. Well organized system testing can improve the reliability of the system and improve user confidence. Thus, the test plan should be designed before the testing step and most of the data should be specified during the design stage. Testing can enforce the module testing, integration testing, and system testing in a stepwise manner.

This development process of an end-user application system may achieve the objectives of database system organizations and may increase simplicity of the system development process for system designers. As a result, users can be convinced that "the database system is best in all systems for personnel management".

VI.   CONCLUSIONS AND RECOMMENDATIONS

The ROK Army personnel management is a complex and time consuming job and needs very accurate information to increase war power. Manual systems can not reduce national defense expenditures and make it difficult to obtain accurate information from all personnel in the Army. Thus, the Army needs a computerized personnel management system. Many file systems can be operated, but a database system is more powerful than the file system approach. Database processing can increase end-user productivity, decrease staff, enable work to be done more efficiently, and permit end-user management more authority and responsibility.

The objectives of database system organizations can be established before the system development begins. Relational database models will be the most useful in the ROK Army's end-user environment, because this model gives structure independency for databases and high level languages for queries. Normal forms and query optimization techniques can be applied to decrease inefficiency of the relational database model in the system design stage. Furthermore, databases which contain the personnel data can be divided into smaller databases by using branch or rank in order to decrease execution time and increase security, if necessary. All data should be collected from military officer personnel

records and regulation books. However, all data can not be collected from personnel records which are currently used in the Army. Thus, the format of personnel records must be changed. The number of databases and the amount of data for each database will depend on the unit size (e.g., the Army Headquarters may require much larger databases than the other units). An end-user application system must be developed, because the end-users who are working in the Army have very little knowledge of database system operations and periodically require statistical information derived from the data. The software engineering goals must be considered, and top-down design methodology and structured programming techniques should be applied as tools. The application system for each unit is dependent on unit size and unit characteristics (i.e., Army H.Q. needs application programs to extract more complex and more aggregated information than the other units in order to increase group effectiveness. On the other hand, divisions need application programs to extract simpler and more individualized personnel information in order to increase individual personnel effectiveness).

Software life cycle considerations guided the development process. This process can be applied to develop database systems for the other departments (G-2, G-3,...). The developed database system can be implemented any way for personnel management for all departments of personnel management in the Army. But, "which implementation method

is more efficient, centralized or decentralized in the Army

environment?" and "what type of hardware should be installed

for each level of unit?" are topics for future research.

APPENDIX A

A BUSINESS MODEL

A business model details how the organization operates and what is required to support the operations. The how and what aspects of an organization can be represented in terms of functions of the organization and the data classes that support these functions.

A function in an organization is an essential activity or decision required to manage the resources and operations of the organization. Functions in an organization are identified by:

(A) Examining statements of purpose of a task or an organizational area.

(B) Examining work programs in an organizational area.

(C) Identifying products or services provided by an organizational area and determining what functions are needed to produce such products or services.

A data class in an organization is an aggregation of data (attributes) that is required by a function or is produced by it. Data classes are identified by examining the data required or produced by a function.

Once the functions and data classes for each function have been identified, the definitions must be examined to assure they are consistent, non-redundant, and clear. A list of functions, data classes, and their definition, as

72

well as a matrix showing which functions use which data

classes, can then be specified. Such a matrix for depart-

ments of ROK Army personnel management is shown on the

following page, and the definition of each function is

given in Chapter II.

data classes	functions					
	procurement	training & education	assignment	treatment	promotion	separation
person	O	O	O	O	O	O
promotion list		O	O		O	O
military carrier list		O	O		O	
military education list		O	O		O	
health condition list			O	O	O	O
punishment/ prize list			O	O	O	O
battle carrier list			O	O	O	
special capability list			O	O	O	
security treatment permission			O			

	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6
foreign language capability list				0	0	
civilian/ committed education list		0		0	0	0
civilian carrier list		0	0	0	0	0
father/ mother			0			0
wife/ child			0			0
dependent family policy			0			
special carrier list	0	0		0		
possible resource in the future	0	0	0	0	0	0
planned military education					0	

assignment policy				O	O	
PX material policy			O			
PX material list purchcased by persons			O			
public service estimation list	O	O		O		
recommended order list for promotion		O				
minimum public service duration policy for promotion		O			O	O
unfitted personnel	O	O		O	O	

76

minimum public service duration policy for retirement	0					0
appointed personnel for promotion/retirement	0	0		0	0	0

APPENDIX B

A SAMPLE LIST OF INTEGRATED ENTITY TYPES

Different users in different functional areas usually require different data in each data class in the Business Model (Appendix A). Thus, common entity types must be extracted from the data classes through the integration of data in all functional areas for each data class. The description which follows names the entity types, defines what they represent, and lists their associated attributes.

1. PERSON

(rank, military serial number, name, social security number, order of son, main address, present address, COMMISSION (name of native military education course, the order of native military education course, DATE of COMMISSION (year, month, day), DATE of BIRTH (year, month, day), DATE of PROMOTION (year, month, day), BRANCH (original, special), FUNCTION (main, secondary), sex, type of blood, type of religion, PUBLIC SERVICE (type of public service, maximum duration of total public service, final year of total public service duration, maximum age for public service, final year of maximum public service age, maximum duration at a pointed rank, final year of maximum duration at a pointed rank).

2. PROMOTION LIST

(military serial number, rank, DATE OF PROMOTION, (year, month, day), type of promotion).

3. MILITARY CARRIER LIST

(military serial number, unit name, unit class, rank, duty name, PERIOD (FROM (year, month, day), UNTIL (year, month, day), number of month)).

4. MILITARY EDUCATION LIST

(military serial number, rank, SCHOOL (name, course, order of course), PERIOD (FROM (year, month, day), UNTIL (year, month, day), number of month), average grade, order of average grade).

5. HEALTH CONDITION LIST

(military serial number, rank, DATE OF CHECKING (year, month, day), CONDITION (EYE (left, right class), EAR (left, right, class), nose class, height, weight, TOOTH (up, down, class), HAND (left, right, class), FOOT (left, right, class), skin class, BLOOD PRESSURE (highest, lowest, class), lung class, neck class, round of chest), result of checking).

6. PRIZE/PUNISHMENT LIST

(rank, military serial number, kind of prize/ punishment, DATE (year, month, day), reason, point for promotion).

7. FOREIGN LANGUAGE CAPABILITY LIST

(military serial number, kind of language, CAPABILITY (speaking, listening, reading, interpretation, translation)).

8. CIVILIAN/(NOT) FINISHED IN(OUT)-COUNTRY COMMITTED EDUCATION LIST

(military serial number, SCHOOL (country, name, address, major, academic degree), PERIOD (FROM (year, month,

day), UNTIL (year, month, day), number of month), graduation classification).

9.   PLANNED MILITARY EDUCATION

        (SCHOOL (name, course order of course), branch, rank, number of person, required course, PERIOD (FROM (year, month, day), UNTIL (year, month, day), number of week)).

10.   ASSIGNMENT POLICY

        (unit class, rank, duty name, branch, function number, number of month for duration, required previous education for duty, required previous duty name for given duty).

11.   FIRST/SECOND SERVICE ESTIMATION LIST

        (rank, military serial number, ESTIMATED DATE (year, month, day), FIRST/SECOND ESTIMATOR (rank, name, duty name), integrity, honesty, responsibility, personality, command capability, average grade, number of total estimatee, order of average grade).

12.   RECOMMENDED ORDER

        (rank, military serial number, RECOMMENDED DATE (year, month, day), RECOMMENDER (rank, name, duty name), number of total recomendee, recommended order).

13.   MINIMUM DURATION POLICY FOR RETIREMENT

        (name of native military education course, type of public service, number of month of the least duration for retirement).

APPENDIX C

A LIST OF THIRD NORMAL FORMS

The original entity types defined in Appendix B have
many anomalies which should be eliminated to realize an
acceptable schema. Anomalies are caused by certain unwanted
functional dependency structures. These anomalies can be
avoided by forcing normal forms concepts and applying decompo-
sition stepwisely. As the result of this effort, a set of
third normal forms are designed and described below.

1.1. PERSON

(rank, military serial number, name, social security
number, order of son, name of native military education
course, the order of native military education course,
main branch, secondary branch, main function, secondary
function, birth year, birth month, birth day, type of blood,
type of religion, type of public service).

*PRIMARY KEY: military serial number

1.2. COMMISSION

(name of native military education course, the order
of native military education course, commissioned year,
commissioned month, commissioned day).

*PRIMARY KEY: name of native military education course

+ the order of native military education course.

1.3. PUBLIC SERVICE LIMITATION POLICY

(rank, maximum duration of total public service, maximum
age for public service, maximum duration at a pointed rank).

*PRIMARY KEY: rank.

1.4. PERSON - 4

(commissioned year, maximum duration of total public service, final year of total public service duration).

*PRIMARY KEY: commissioned year + maximum duration of total public service.

*This subschema can be manipulated from (1-2) and (1-3) using a DBMS.

1.5. PERSON - 5

(birth year, maximum age for public service, final year of maximum public service age)

*PRIMARY KEY: birth year + maximum age for public service.

*This subschema can be manipulated from (1-1) and (1-3) using a DBMS.

1.6. PERSON - 6

(promoted year, maximum duration at a pointed rank, final year of maximum duration at a pointed rank).

*PRIMARY KEY: promoted year + maximum duration at a pointed rank.

*This subschema can be manipulated from (1-3) and (2) using a DBMS.

2. PROMOTION LIST

*Same as (2) in Appendix B, but combined attributes must be converted into simple attributes.

*PRIMARY KEY: rank + military serial number.

3. MILITARY CARRIER LIST

*Same as (3) in Appendix B, but combined attributes must be converted into simple attributes.

*PRIMARY KEY: military serial number + rank + duty name

                + name of unit.

   4.   MILITARY EDUCATION LIST

*Same as (4) in Appendix B, but combined attributes must be

converted into simple attributes.

*PRIMARY KEY: military serial number + rank + school name +

                course name + order of course.

   5.   HEALTH CONDITION LIST

*Same as (5) in Appendix B, but combined attributes must be

converted into simple attributes.

*PRIMARY KEY: military serial number + year.

   6.1. PRIZE/PUNISHMENT LIST

         (rank, military serial number, kind of prize/punishment,

received year, received month, received day, reason).

*PRIMARY KEY: military serial number + kind of prize +

                received year + received month.

   6.2. PRIZE POINT

         (kind of prize/punishment, point for promotion)

*PRIMARY KEY: kind of prize.

   7.   FOREING LANGUAGE CAPABILITY LIST

*Same as (7) in Appendix B, but combined attributes must

be converted into simple attributes.

*PRIMARY KEY: military serial number + kind of language.

   8.1. CIVILIAN/(NOT) FINISHED IN (OUT)-COUNTRY COMMITTED
         EDUCATION LIST

         (military serial number, school name, major, academic

degree, from year, from month, from day, until year, until

month, until day, number of month, graduation classification).

*PRIMARY KEY: military serial number + school name + major + academic degree.

8.2. IN (OUT)-COUNTRY SCHOOL ADDRESS

(school name, (country name,) school address)

*PRIMARY KEY: school name.

9.1. PLANNED MILITARY EDUCATION

(school name, course name, order of course, number of person, from year, from month, from day, until year, until month, until day)

*PRIMARY KEY: school name + course name + order of course.

9.2. MILITARY EDUCATION POLICY

(course name, rank, required course, number of week)

*PRIMARY KEY: course name.

9.3. MILITARY EDUCATION COURSE

(school name, course name, branch)

*PRIMARY KEY: school name + course name.

10. ASSIGNMENT POLICY

*Same as (10) in Appendix B.

*PRIMARY KEY: unit classification + rank + duty name.

11. FIRST/SECOND SERVICE ESTIMATION LIST

*Same as (11) in Appendix B, but combined attributes must be converted into simple attributes.

*PRIMARY KEY: military serial number + estimated year.

APPENDIX D

A SAMPLE LIST OF RELATIONSHIPS

A relationship corresponds to an aggregation of two or more entity types. For each entity type, a description of each relationship containing its name, entity types related, and mapping (one to one, one to many and many to many) is produced. For example, PERSON HAS BEEN COMMISSIONED can be represented as an aggregation of the entity types PERSON and COMMISSION. A relationship is binary when only two entity types are aggregated, or of higher order (e.g., n-ary). However, most DBMSs handle only binary relationships.

(e.g.) PERSON HAS BEEN COMMISSIONED (relationship)

       : between PERSON and COMMISSION

       : one to one (mapping)

1. PERSON HAS BEEN PROMOTED

       : between PERSON and PROMOTION LIST

       : one to many

2. PERSON HAS CARRIED MILITARY CARRIER

       : between PERSON and MILITARY CARRIER LIST

       : one to many

3. PERSON HAS STUDIED MILITARY EDUCATION

       : between PERSON and MILITARY EDUCATION LIST

       : one to many

APPENDIX E

A SAMPLE DATA DICTIONARY

A data dictionary is primarily a dictionary that defines the internally necessary attributes of the data for each database. The main functions of the data dictionary are as follows.

1. The dictionary highlights ambiguity and inconsistency in the data. Standard names, constraints, and sources are established.

2. The dictionary provides documentation of all organizational data. It is a guide to what data is available, what it is called, where it is used, etc.

3. The dictionary assists the DBA in maintaing configuration control over database.

The data dictionary should contain access authorities. These are contained in each database. To apply the above functions, the dictionary below contains 6 columns. These are:

1. Field Name - This column indicates the standard field name to represent attributes.

2. Explanation - This column explains the potential values for a given field.

3. Type - This column represents a type of field value where "n" means numeric, "an" means alphanumeric and "a" means alphabetic.

4.   Length - This column represents the number of digits for a potential field value.

5.   Limited value - This field restricts the potential values for a given field if the number of field values is limited.

6.   Remark - This column is free format, and includes keys and comments.

However, the number of digits for field names as variables, types of fields, and length of field values are determined by DBMS capabilities and data characteristics.

1. PERSON.DB. (database for basic personnel data) (access authority: write-record officer, read-all)

field name	explanation	type	length	limited values	remarks
rank	rank of person	an	3	1LT, 2LT,...	secondary key
msn	military serial number	n	7		primary key
name	name of person	a	15		last, first, middle
ssn	social security number	n	9		secondary key
orson	order of son	n	1	0,........9	woman : Ø
maadd	main address	an	60		
pradd	present address	an	60		
nnmec	name of native military education course	an	5	kma,rotc, 3ma,...	
onmec	order of native military education	n	3		
mainbr	main branch	a	4	infa, armo,...	secondary key
secobr	secondary branch	a	4	insp, info,...	
byear	birth year	n	4	1920.....1965	secondary key

field name	explanation	type	length	limited values	remarks
mainfu	main function	n	3	100,....,999	
secofu	secondary function	n	3		
bmonth	birth month	n	2	1,........12	
bday	birth day	n	2	1,........31	
btype	blood type	a	2	a, b, ab, 0	
rtype	type of religion	a	4	cato,....,none	
ptype	type of public service	a	1	L, s	L:long, s:short

2. PSLPOL.DE (database for public service limitation policy) (access authority: write-policy officer, read-all)

field name	explanation	type	length	limited values	remarks
rank	military rank	an	3	1LT, 2LT,.....	primary key
maxtot	maximum duration of total public service	n	2	4,.........35	unit:year
maxage	maximum age for public service	n	2	30,......60	unit:age
maxran	maximum duration at a pointed rank	n	2	1,........12	unit:year

APPENDIX F

A SAMPLE LIST OF TRANSACTION PROCESSING ANALYSIS

The analysis of transaction processing specifies inputs and outputs, but does not involve the specific steps to obtain the output. This analysis is very helpful to end-user application system designers because it affects the selection of levels and modules in the baseline hierarchical diagram of an application system. This analysis may result in a documentation as illustrated below.

1. List of all officers' highest academic degree who are commissioned in a specific year for each source organization.

    a.   functional areas : personnel policy department.

    b.   kind of access : retrieval.

    c.   entity types : commission, civilian education, person.

    d.   relationships : person has been commissioned, person has studied civilian education.

    e.   frequency : each year.

    f.   Processing priority : III (priority could be divided into 3 types: I, II and III).

    g.   security : limited access.

    h.   report formats.

        (1)   source organization : xxxxxxxxxxx :

            (a)   PH.D.

rank	msn	name	school name	major

(2)   MS : same format

             (3)   MA : same format
                        :
                        :
             (7)   none (high school) : same format

    b.   source of organization : same format.
                    :
                    :
    g.   source of organization : same format.

  2.   List of all officers who have performed some specific
duty and received military education to select some officers
for new assignments.

    a.   functional areas : assignment department.

    b.   kind of access : retrieval.

    c.   entity types : person, military carrier, list,
         military education list.

    d.   relationship types : person has military carrier,
         person has studied military education.

    e.   frequency : every day.

    f.   access priority : II.

    g.   security : free.

    h.   report format.

rank	msn	name	unit	duty position

        Total : xxxxxxxx :

LIST OF REFERENCES

1. Bohem, B. W., "Software and Its Impact; A Quantitative Assessment", _Datamation_, vol. 19, No. 5, May 1973.

2. McMclure, C. L., _Managing Software Development and Maintenance_, Van Nostrand Reinhold Company, 1981.

3. _Korean Army Personnel Management Regulations_, HQ of Army, 1980.

4. _Korean Army Unit Education and Training Regulations_, HQ of Army, 1980.

5. Dunnette, Marvin D., _Personnel Selection and Placement_, Belmonte: Wandsworth Publishing Company Inc., 1966.

6. _The Military Public Service Regulations_, Department of National Defense, 1980.

7. Martin, J., _Computer Data-Base Organization_, Prentice-Hall Inc., Englewood Cliffs, N.J., 1977.

8. Kronke, D., _Database Processing_, Science Research Associates Inc., Chicago, Tronto, 1977.

9. Ullman, J.D., _Principle of Database System_, Computer Science Press, Inc., Rockville, Maryland, 1980.

10. Date, C. J., _An Introduction to Database Systems_, Addison-Wesley Publishing Company Inc., 1981.

11. Tsichritzis, D. C. and Lockovsky, F. H., _Data Models_, Prentice-Hall Inc., Englewood Cliffs, N.J., 1982.

12. Biggs, C. L., Birks, E. G. and Atkins, W., _Managing the System Development Process_, Prentice-Hall Inc., Englewood Cliffs, N.J., 1980.

13. Zelkowitz, M. V., "Perspectives Software Engineering", _Computing Surveys_, vol. 10, No. 2, June 1978.

14. Dijkstra, E. W., "Go to Statement Considered Harmful", _Communications_ of the ACM, vol. 11, No. 3, March 1968.

15. Wulf, W. A., "A Case Against the Goto", <u>Communications of the ACM</u>, 1972.

16. Ramamoorhy, C. V., and Ho, S. F., "Testing Large Software with Automated Software Evaluation System", <u>Current Trends in Programming Methodology</u>, vol. II, Prentice-Hall Inc., Englewood Cliffs, N.J., 1982.

17. Ross, D. T., and Goodenough, J. B., "Software Engineering: Process, Principles, and Goals", <u>Computer</u>, May 1975.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center   2
    Cameron Station
    Alexandria, Virginia 22314

2. Library, Code 0142   2
    Naval Postgraduate School
    Monterey, California 93940

3. Department Chairman, Code 52   2
    Department of Computer Science
    Naval Postgraduate School
    Monterey, California 93940

4. Professor S. H. Parry, Code 55Py   2
    Department of Operations Research
    Naval Postgraduate School
    Monterey, California 93940

5. Professor R. S. Elster, Code 54Ea   1
    Department of Administrative Science
    Naval Postgraduate School
    Monterey, California 93940

6. Professor D. Z. Badal, Code 52Zd   1
    Department of Computer Science
    Naval Postgraduate School
    Monterey, California 93940

7. Central Computer Center   2
    Army Headquarters, 140-01
    Seoul, Republic of Korea

8. Professor Yoon.Ki.Chae   1
    Department of Mathematics
    Liberal Arts and Sciences College
    Kyung Pook National University, Daegu
    Seoul, Republic of Korea

9. Dae Sik Hong   10
    346 Yangji-li Sinwoen-myun
    Keochang-gun Kyungnam-do, 640-43
    Seoul, Republic of Korea

10. Chief of Staff   1
     Army Headquarters, 140-01
     Seoul, Republic of Korea

11. Division of Computer Management   2
     Department of Management
     Army Headquarters, 140-01
     Seoul, Republic of Korea

12. Department of Personnel Management   1
     Army Headquarters, 140-01
     Seoul, Republic of Korea

13. Division of Education   1
     Department of Personnel Management
     Army Headquarters, 140-01
     Seoul, Republic of Korea

14. CDR J. Pfeiffer   1
     Computer Technology Curriculum, Code 37
     Naval Postgraduate School
     Monterey, California 93940